Learning-Based Constraint Satisfaction With Sensing Restrictions

Alessandro Checco, Douglas J. Leith
Hamilton Institute, NUI Maynooth*

November 26, 2012

Abstract

In this paper we consider graph-coloring problems, an important subset of general constraint satisfaction problems that arise in wireless resource allocation. We constructively establish the existence of fully decentralized learning-based algorithms that are able to find a proper coloring even in the presence of strong sensing restrictions, in particular sensing asymmetry of the type encountered when hidden terminals are present. Our main analytic contribution is to establish sufficient conditions on the sensing behaviour to ensure that the solvers find satisfying assignments with probability one. These conditions take the form of connectivity requirements on the induced sensing graph. These requirements are mild, and we demonstrate that they are commonly satisfied in wireless allocation tasks. We argue that our results are of considerable practical importance in view of the prevalence of both communication and sensing restrictions in wireless resource allocation problems. The class of algorithms analysed here requires no message-passing whatsoever between wireless devices, and we show that they continue to perform well even when devices are only able to carry out constrained sensing of the surrounding radio environment.

1 Introduction

Many fundamental wireless network allocation tasks can be formulated as constraint satisfaction problems, including channel and sub-carrier allocation [8], TDMA scheduling [4, 9], scrambling code allocation [5], network coding [8] and so on. Importantly, these tasks must often be solved while respecting strong communication constraints due, for example, to the range over which devices can communicate being smaller than the range over which they interfere or otherwise interact. Recently, fully decentralised Communication-Free Learning (CFL) algorithms have been proposed for solving general constraint satisfaction problems without the need for message-passing [8]. These CFL algorithms

*Work supported by Science Foundation Ireland grant 11/PI/11771.

exploit local sensing to infer satisfaction/dissatisfaction of constraints, thereby avoiding the need for message-passing and use stochastic learning to converge to a satisfying assignment with probability one. Convergence of these CFL algorithms to a satisfying assignment is, however, only guaranteed when *all* devices participating in a constraint are able to sense the satisfaction/dissatisfaction of the constraint. This sensing requirement is violated in a number of important practical problems, for example in wireless networks with hidden terminals. The main contribution of the present paper is a new analysis of CFL-like algorithms which establishes that much weaker requirements on sensing are sufficient to guarantee convergence to a solution. The analysis of stochastic learning algorithms is challenging and part of the technical contribution is the development of novel analysis tools. We present a number of examples demonstrating the efficacy of CFL-like algorithms when subject to strong sensing as well as communication constraints, and explore the impact of sensing constraints on the rate of convergence.

A Constraint Satisfaction Problem (CSP) consists of N variables, $\vec{x} := (x_1, \dots, x_N)$, and M clauses, i.e. $\{0,1\}$-valued functions, $(\phi_1(\vec{x}), \dots, \phi_N(\vec{x}))$. An assignment $\vec{x}$ is a solution if all clauses simultaneously evaluate to 1. In problems derived from network applications, each constrained variable x_i is often associated to a physically distinct device, such an access point or a base-station. For example,consider a collection of WLANs operating in an unlicensed radio band. Each WLAN can choose one of several channels to operate on and the WLANs require to jointly select channels so as to avoid excessive interference between the WLANs. We can formulate this task as a CSP by letting x_i be the channel selected by WLAN $i \in \{1, \dots, N\}$ and defining $M = N(N-2)/2$ clauses, one for each pair of WLANs which evaluates to one if the WLANs are non-interfering, or are out of interference range, and evaluates to zero otherwise. Communication between the devices is impeded by multiple causes: the interference range of a typical wireless device is considerably larger than its communication range, and thus WLANs can interfere but may be unable to communicate; WLANs can have different administrative domains that would prevent communication even via a wired backhaul and even if a proper knowledge of the physical location of the different WLANs was known. Consequently, the selection of the variables x_i in a distributed manner (allowing message passing) has to be discarded, mandating a fully decentralized channel-selection algorithm.

A practical CSP solver for this task can only rely on each WLAN being able to measure whether or not its current choice of channel is subject to excessive interference. Importantly, observe that this sensing need not be symmetric. The scenario in Figure 1 illustrates this feature: here transmissions on link $A - B$ interfere with transmissions on link $C - D$, but not vice versa *i. e.* transmitter A acts as a hidden terminal affecting link $C - D$. Such asymmetry in sensing is ubiquitous in networks with hidden terminals. The analysis in [8] requires that all links sharing a channel are able to sense whether any one or more of the links is experiencing excessive interference and so dissatisfied, and therefore is not applicable to networks with hidden terminals.

In the present paper our aim is to address this deficiency. We focus on graph-

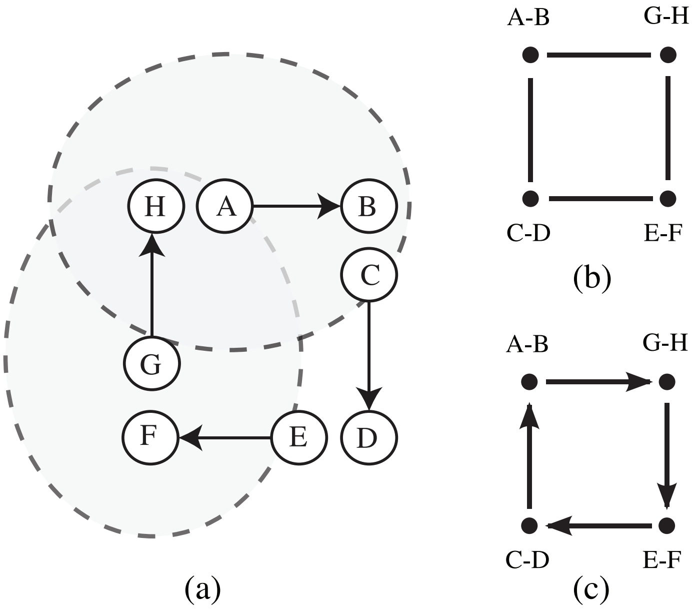

Figure 1: (a) Illustrating a wireless network with asymmetric sensing due to hidden terminals. The shaded areas indicate the interference created by transmitters A and G. Transmissions by A prevent H receiving transmissions by G. However, the converse is not true *i.e.* transmissions by G do not prevent B from successfully receiving transmissions by A. Link $A - B$ can therefore be satisfied while $G - H$ is dissatisfied. Similarly for the other links shown. Associating each edge with a vertex yields graph (b) corresponding to (a) for which a proper coloring is sought. Sensing restrictions then yield the induced oriented graph (c), as explained in Section 4.1.

coloring problems, a subset of general CSPs, and constructively establish the existence of decentralized learning-based solvers that are able to find satisfying assignments even in the presence of sensing asymmetry. We establish sufficient conditions on the sensing behaviour to ensure that the solvers find satisfying assignments with probability one. We demonstrate that these conditions are commonly satisfied in wireless allocation tasks and explore the impact of sensing constraints on the speed which a satisfying assignment is found.

Even if in certain settings a limited amount of communication between the devices may be possible, like for example overhearing beacons of traffic from some of the interferers, this information is topology dependent and cannot be assumed during the design of the algorithm. The opportunistic exploitation of such partial information is left for future work.

2 Related Work

The graph coloring problem has been the subject of a vast literature, from cellular networks (e.g. [22]), wireless LANs (e.g. [22, 20, 19, 18, 21] and references therein) and graph theory (e.g. [7, 16, 12, 13]). Almost all previous work has been concerned either with centralised schemes or with distributed schemes that employ extensive message-passing. Centralised and message-passing schemes have many inherent advantages. In certain situations, however, these systems may not be applicable. For example, differing administrative domains may be present in a network of WLANs.

An exception is the work of Kauffmann et al. [15, 14], which proposes a distributed simulated annealing algorithm for joint channel selection and association control in 802.11 WLANs. However, heuristics are used to both terminate the algorithm and to restart it if the network topology changes. Network-wide stopping/restarting in a distributed context can be challenging without some form of message-passing.

In the field of graph theory, Dousse [7], Hedetniemi et al. [12], Johansson [13] address the problem of graph coloring, when the amount of colors available is large (typically $\delta+1$) and allowing some form of message passing in an undirected graph. The only exception is in [16], where a sort of directionality of the graph is considered: the distributed nodes can make a choice in a hierarchical manner, *i. e.* when $i \to j$, node i may keep its choice even if j has same color, but this is made possible assuming the existence of this hierarchy is known and that there is still a bidirectional channel available for communication.

This work builds upon the works of the early work of Clifford and Leith [6], then refined and extended by Leith et al. [17], Duffy et al. [8]; they present a fully decentralised CFL, proven to solve a large class of problems that include graph coloring, but without sensing restrictions.

3 Preliminaries

We will now introduce the problem, using same notation of [8], extending it to encompass sensing restriction in section 4.1.

3.1 Coloring Problems (CPs)

Let $G = (V, \mathcal{M})$ denote an undirected graph with set of vertices $V = \{1, \ldots, N\}$ and set of edges $\mathcal{M} := \{(i,j) : i,j \in V, i \leftrightarrow j\}$, where $i \leftrightarrow j$ denotes the existence of an edge joining vertices $i, j \in V$. Note that with this notation the edges in set $\mathcal{M}$ are directed, since this will prove convenient later when considering oriented subgraphs of G. However, since graph G is undirected we have $(i,j) \in \mathcal{M} \iff (j,i) \in \mathcal{M}$.

A coloring problem (CP) on graph G with $D \in \mathbb{N}$ colors is defined as follows. Let $x_i \in \mathcal{D}$ denote the color of vertex i, where $\mathcal{D} = \{1, \ldots, D\}$ is the set of available colors, and $\vec{x}$ denote the vector $(x_1, \ldots, x_N)$. Define clause $\Phi_m \colon \mathcal{D}^N \mapsto \{0,1\}$ for each edge $m = (i,j) \in \mathcal{M}$ with:

$$\Phi_m(\vec{x}) = \Phi_m(x_i, x_j) = \begin{cases} 1 & \text{if } x_i \neq x_j \\ 0 & \text{otherwise} \end{cases}.$$

We say clause $\Phi_m(\vec{x})$ is *satisfied* if $\Phi_m(\vec{x}) = 1$. An assignment $\vec{x}$ is said to be satisfying if for all clauses $m \in \mathcal{M}$ we have $\Phi_m(\vec{x}) = 1$. That is

$$\vec{x} \text{ is a satisfying assignment iff } \min_{m \in \mathcal{M}} \Phi(m) = 1. \tag{1}$$

Equivalently, $\vec{x}$ is a satisfying assignment if and only if $x_i \neq x_j$ for all edges $(i,j) \in \mathcal{M}$ *i.e.* if $i \leftrightarrow j$. A satisfying assignment for a coloring problem is also called a *proper coloring*.

Definition 1 (Chromatic Number)**.** The chromatic number $\chi(G)$ of graph G is the smallest number of colors such that at least one proper coloring of G exists. That is, we require the number of colors D in our palette to be greater than $\chi(G)$ for a satisfying assignment to exist.

3.2 Decentralized CP Solvers

Definition 2 (CP solver)**.** Given a CP, a CP solver realize a sequence of vectors $\{\vec{x}(t)\}$ such that for any CP that has a satisfying assignment

(D1) for all t sufficiently large $\vec{x}(t) = \vec{x}$ for some satisfying assignment $\vec{x}$;

(D2) if t' is the first entry in the sequence $\{\vec{x}(t)\}$ such that $\vec{x}(t')$ is a satisfying assignment, then $\vec{x}(t) = \vec{x}(t')$ for all $t > t'$.

In order to give criteria for classification of decentralized CP solvers, we re-write the LHS of Equation (1) to focus on the satisfaction of each variable

$$\vec{x} \text{ is a satisfying assignment iff } \min_{i \in V} \min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}) = 1. \tag{2}$$

where $\mathcal{M}_i$ consists of all edges in $\mathcal{M}$ that contain vertex i, *i. e.*

$$\mathcal{M}_i = \{(j,i) \colon (j,i) \in \mathcal{M}\}\,.$$

Note that we adopt the convention of including edges in $\mathcal{M}_i$ which are incoming to vertex i, but since $(i,j) \in \mathcal{M} \iff (j,i) \in \mathcal{M}$ then $\bigcup_{i \in V} \mathcal{M}_i = \mathcal{M}$.

A decentralized CP solver is equivalent to a parallel solver, where each variable x_i run independently an instance of the solver, having only the information on whether all the clauses that x_i participates in are satisfied or at least one clause is unsatisfied. The solver located on variable x_i must make its decisions only relying on this information.

Definition 3 (Decentralized CP solver)**.** A decentralized CP solver is a CP solver that for each variable x_i, must select its next value based only on the evaluation of

$$\min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}). \tag{3}$$

That is, the decision is made without knowing

(D3) the assignment of x_j for $j \neq i$.

(D4) the set of functions that any variable, including itself, participates in, $\mathcal{M}_j$ for $j \in V$.

(D5) the functions Φ_m for $m \in \mathcal{M}$.

4 Coloring Problems With Sensing Restrictions

4.1 Decentralised Solvers

Sensing restrictions mean, for example, that a hidden terminal is unable to sense whether or not its transmissions are causing excessive interference to the set of receivers for which it is hidden. In other words, variable x_i can only evaluate $\min_{m \in \mathcal{C}_i} \Phi_m(\vec{x})$ rather than $\min_{m \in \mathcal{M}_i} \Phi_m(\vec{x})$, where $\mathcal{C}_i \subset \mathcal{M}_i$.

Definition 4 (Decentralized CP Solver With Sensing Restrictions)**.** A Decentralised CP solver where (3) is replaced with the restriction that for each variable x_i, must select its next value based only on an evaluation of

(D6) $\min\limits_{m \in \mathcal{C}_i} \Phi_m(\vec{x})$, where *information set* $\mathcal{C}_i \subseteq \mathcal{M}_i$ is a subset of edges incoming to node i and we adopt the convention that $\min\limits_{m \in \emptyset} \Phi_m(\vec{x}) = 1$.

Note that despite the sensing restrictions we still require the solver to satisfy $(D1)$ and find a satisfying assignment, *i. e.* for all t sufficiently large $\vec{x}(t) = \vec{x}$ with $\min_{i \in V} \min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}) = 1$.

It is important to note that an assignment $\vec{x}$ may ensure $\min_{m \in \mathcal{C}_i} \Phi_m(\vec{x}) = 1$, $i \in V$ but might have $\min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}) = 0$ for one or more variables and so need not be satisfying in the absence of sensing restrictions *i. e.* it may not be a proper coloring. We therefore require the following sensing condition in order to satisfy $(D1)$:

Lemma 1. Let $\mathcal{C} := \cup_{i \in V} \mathcal{C}_i$. Suppose at least one half of each undirected edge $i \leftrightarrow j$ in $\mathcal{M}$ appears in at least one information set C_i for some vertex i *i. e.* $(i,j) \in \mathcal{M} \implies (i,j)$ or $(j,i) \in \mathcal{C}$. Then an assignment $\vec{x}$ is satisfying with sensing restrictions iff it is satisfying in the absence of sensing restrictions. That is, $\min_{i \in V} \min_{m \in \mathcal{C}_i} \Phi_m(\vec{x}) = 1 \iff \min_{i \in V} \min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}) = 1$.

Proof. Suppose $\min_{i \in V} \min_{m \in \mathcal{C}_i} \Phi_m(\vec{x}) = 1$. That is, $\Phi_m(\vec{x}) = 1 \ \forall m \in \mathcal{C}$. By definition, $\Phi_{(i,j)}(\vec{x}) = \Phi_{(j,i)}(\vec{x})$ and since $(i,j) \in \mathcal{M} \implies (i,j)$ or $(j,i) \in \mathcal{C}$ the result follows. Conversely, suppose $\min_{i \in V} \min_{m \in \mathcal{M}_i} \Phi_m(\vec{x}) = 1$. Since $\mathcal{C}_i \subseteq \mathcal{M}_i$, the result immediately follows. □

It will prove useful to consider oriented partial graph $G' = (V, \mathcal{C})$ induced by the information set $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. This graph has the same set V of vertices as graph G for which a proper coloring is sought, but the edges are now defined by the set of *ordered* pairs $(i,j) \in \mathcal{C} \quad \text{if } (i,j) \in \mathcal{C}_j$. We say $i \to j$ if there is a directed edge from i to j, and $i \not\to j$ if there is no edge from i to j. For example, Figure 1(c) gives the graph G' corresponding to Figure 1(b). Here, the directed edge from $A - B$ to $G - H$ indicates that while $G - H$ can sense whether the edge between $A - B$ and $G - H$ is satisfied or not, $A - B$ cannot.

4.2 Examples

Before proceeding, we briefly demonstrate that several important resource allocation tasks in wireless networks fall within our framework of graph coloring with sensing restrictions.

4.2.1 Channel Allocation With Hidden Terminals

Consider a network of N wireless links $i = 1, \ldots, N$, each consisting of a transmitter T_i and a receiver R_i. Let P_i denote the transmit power of T_i and γ_{ij} denote the path loss between the transmitter T_i of link i and the receiver R_j of link j. The received power at R_i from T_j is therefore $\gamma_{ji} P_j$. Each link can select one from a set $\mathcal{D} = \{1, \ldots, N\}$ of available channels to use. Link i would like to select a channel in such a way that the signal power impinging on the receiver R_i from other links sharing the same channel is less than a specified threshold Q_i – Q_i may, for example, be selected to ensure that the SINR at R_i is above a target threshold. Each link i can sense that another link j is sharing the same channel when the received power $\gamma_{ji} P_j \geq Q_j$ (this might correspond to the minimum interference power that causes decoding errors on the link or to the carrier-sense threshold in 802.11).

To formulate this as a coloring problem, let $\mathcal{D} = \{1, \ldots, N\}$ be the palette of available colors. Associate variable x_i with wireless link i, $i \in \{1, \ldots, N\}$, with the value of $x_i \in \mathcal{D}$ corresponding to the channel selected by link i. Define graph $G = (V, \mathcal{M})$ with $V := \{1, \ldots, N\}$ and set of edges $\mathcal{M}$. Add edge (j, i) to $\mathcal{M}$ whenever the received power $\gamma_{ji} P_j$ from link j at link i is above threshold Q_i when both links select the same channel, $i \neq j$, $i, j \in \{1, \ldots, N\}$. Importantly, whenever an edge (j, i) is in $\mathcal{M}$ we also add edge (i, j) to $\mathcal{M}$, so that G is an undirected graph. A proper coloring of graph G corresponds to a satisfactory channel allocation *i. e.* $\gamma_{ji} P_j > Q_i$, for all $i \in \{1, \ldots, N\}$ and all j such that $x_j = x_i$ and $j \in \{1, \ldots, N\}$.

Now define graph $G' = (V, \mathcal{C})$ with edge $(j, i) \in \mathcal{C}$ when the received power $\gamma_{ji} P_j$ from link j at link i is above threshold Q_i when both links select the same channel. Note that, unlike for graph G, we do *not* also add edge (i, j) to $\mathcal{C}$ unless $\gamma_{ij} P_i > Q_j$ when both links select the same channel. Observe that the edges in graph G' embody the sensing abilities of links, and in general $\mathcal{C} \neq \mathcal{M}$ and so $G' \neq G$.

Note that we can readily generalise this formulation to include, for example, the selection of multiple channels/sub-carriers by each link and to allow multiple transmitters and receivers in a link (which might then correspond to a WLAN).

4.2.2 Decentralised TDMA Scheduling With Hidden Terminals

When using a time division access scheme, wireless networks need to have a schedule for accessing the channel. This schedule can be decided in a centralized manner using TDMA, but it is possible to require a decentralized way of solving the problem. The classical CSMA/CA approach to decentralized scheduling does not guarantee the convergence to a single schedule and so it can lead to continual collisions. Recently, there has been interest in decentralized approaches for finding collision-free schedules [9]. Consider a wireless network with N links, $i = 1, \ldots, N$. Time is slotted and partitioned into periodic schedules on length $T \geq N$ slots. The transmitter on each link would like to select a slot that is different from the choice made by other transmitters if their collisions would collide (transmissions in the same slot need not collide when, for example, the two transmitters are located sufficiently far apart). A link is able to sense whether its transmission in a slot was successful or not.

To formulate this as a coloring problem, let $\mathcal{D} = \{1, \ldots, N\}$ be the set of available time slots in the periodic schedule. Associate variable x_i with link i, $i \in \{1, \ldots, N\}$, with the value of $x_i \in \mathcal{D}$ corresponding to the slot selected by the transmitter of link i. Define graph $G = (V, \mathcal{M})$ with $V := \{1, \ldots, N\}$ and set of edges $\mathcal{M}$. Add edge (j, i) to $\mathcal{M}$ whenever simultaneous transmissions by the transmitters of links i and j would lead to failure of the transmission by i. Whenever an edge (j, i) is in $\mathcal{M}$, also add edge (i, j) to $\mathcal{M}$. A proper coloring of graph G corresponds to a non-colliding schedule.

Define graph $G' = (V, \mathcal{C})$ with edge $(j, i) \in \mathcal{C}$ when simultaneous transmissions by the transmitters of links i and j would lead to failure of the transmission by j. Unlike for graph G, we do not also add edge (i, j) to $\mathcal{C}$ unless simultaneous transmissions by transmitters i and j would lead to failure of the transmission by j. Once again, the edges in graph G' embody the sensing abilities of links and in general $\mathcal{C} \neq \mathcal{M}$.

5 Solving Coloring Problems With Sensing Restrictions

5.1 Algorithm

Consider stochastic learning Algorithm 1, an instance of this algorithm being run in parallel for every variable.

Algorithm 1

```
Initialize $p_{i,j} = 1/D, j \in \{1, \ldots, D\}$.
loop
    Realize a random variable, selecting $x_i = j$ with probability $p_{i,j}$.
    Evaluate $\min_{m \in \mathcal{C}_i} \Phi_m(\vec{x})$, returning satisfied if its value is 1, and unsatisfied otherwise.
    Update: If satisfied,
        $p_{i,j} = \begin{cases} 1 & \text{if } j = x_i \\ 0 & \text{otherwise.} \end{cases}$
    If unsatisfied,
        $p_{i,j} = \begin{cases} (1-b)p_{i,j} + a/(D-1+a/b) & \text{if } j = x_i \\ (1-b)p_{i,j} + b/(D-1+a/b) & \text{otherwise,} \end{cases}$
    where $a, b \in (0, 1]$ are design parameters.
end loop
```

Algorithm 1 contains design parameters a, $b \in (0, 1)$. In the examples in this paper we select $a = b = 0.1$, and do not optimise these values to particular settings.

In order to be a decentralised CP solver with sensing restrictions, Algorithm 1 must satisfy conditions $(D1) - (D6)$. We can see immediately that Algorithm 1 satisfies $(D2) - (D6)$.

(D3)-(D6) By construction, the only information used by the algorithm is $\min_{m \in \mathcal{C}_i} \Phi_m(\vec{x})$ in Step 4 and thus it satisfies the criteria (D3), (D4), (D5) and (D6).

(D2) Algorithm 1 also satisfies the (D2) criterion that it sticks with a solution from the first time one is found. To see this, note that the effect of Step 5 is that if a variable experiences success in all functions Φ that it participates in it continues to select the same value with probability 1. Thus if all variables are simultaneously satisfied in all clauses, *i. e.* if $\min_{m \in \mathcal{C}_i} \Phi_m(\vec{x})$, then the same assignment will be reselected indefinitely with probability 1.

It remains to verify satisfaction of $(D1)$, *i. e.* convergence of the algorithm to a satisfying assignment, which is the subject of the next section.

5.2 Convergence Analysis

Recall the following definition:

Definition 5 (Strongly Connected Graph)**.** A path of length q in oriented graph $G' = (V, \mathcal{C})$ is a sequence $\mu = (u_1, u_2, \ldots, u_q)$ of edges in $\mathcal{C}$ such that the terminal endpoint of edge u_i is the initial endpoint of edge u_{i+1} for all $i < q$. Oriented graph $G' = (V, \mathcal{C})$ is strongly connected if it contains a path starting in x and ending in y, for each pair of distinct vertices $x \neq y \in V$.

We now state our main analytic result:

Theorem 1. *Consider any satisfiable coloring problem with graph $G = (V, \mathcal{M})$ and information sets $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$. Suppose:*

(A) *At least one half of each undirected edge $i \leftrightarrow j$ in $\mathcal{M}$ appears in at least one information set C_i for some vertex i,* i. e. $(i, j) \in \mathcal{M} \implies (i, j)$ *or* $(j, i) \in \mathcal{C}$*;*

(B) *The induced graph $G' = (V, \mathcal{C})$ is strongly connected.*

Then with probability greater than $1 - \epsilon \in (0, 1)$, the number of iterations for Algorithm 1 to find a satisfying assignment is less than

$$(N^3) \exp(N^4 \log(\gamma^{-1})) \log(\epsilon^{-1}) \text{ where } \gamma = \frac{\min(a, b)}{D - 1 + a/b}.$$

Proof. See Appendix. □

As Theorem 1 covers any arbitrary CP that admits a solution, for any given instance these bounds are likely to be loose. They do, however, allow us to conclude the following corollary proving that if a solution exists, Algorithm 1 will almost surely find it:

Corollary 1. For any coloring problem that admits a proper coloring and that fulfills conditions (A) and (B), Algorithm 1 will find a proper coloring in almost surely finite time.

Intuitively, we expect that sensing restrictions may increase the time it takes to find a satisfying assignment. When $\mathcal{C}_i = \mathcal{M}_i$, $i \in V$ (perfect sensing) then $\mathcal{C} = \mathcal{M}$ and our analysis yields the following bound on the convergence rate:

Corollary 2. When $\mathcal{C}_i = \mathcal{M}_i \forall i \in V$, with probability greater than $1 - \epsilon \in (0, 1)$, the number of iterations for Algorithm 1 to find a satisfying assignment is less than

$$(N) \exp(\frac{N(N+1)}{2} \log(\gamma^{-1})) \log(\epsilon^{-1}),$$

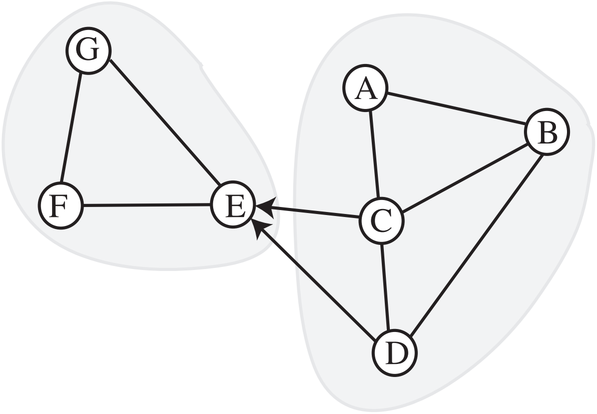

Figure 2: Example of a graph G with two strongly connected components ($\{A, B, C, D\}$ and $\{E, F, G\}$) which are sparsely interconnected. The chromatic number $\chi(G)$ of graph G is 4, the chromatic numbers of the connected components are 3 and 2 respectively.

Proof. See Appendix. □

That is, our upper bound on convergence rate is improved from N^4 to N^2 with perfect sensing. This corresponds to the bound found in [8] for generic DCS problems, but it is looser than the refined bound found there for graph coloring problems. However, it is important to stress that this observation comes with the caveat that, as already noted, we believe both of these bounds are extremely loose. Hence, we revisit this question below using numerical simulations, which yield tight measurements of convergence rate.

5.3 Relaxing Strong Connectivity Requirement

The requirement in Theorem 1 for the sensing graph G' to be strongly connected can be relaxed in a number of ways. If graph G is not connected, we only have to ask for strong connectivity separately for the induced graph corresponding to each connected component. More generally, we can extend our analysis to situations where graph G consists of a number of strongly connected components with sufficiently sparse interconnections between these components.

To help gain insight, consider the example graph G shown in Figure 2. Graph G consists of two strongly connected components, $\{A, B, C, D\}$ and $\{E, F, G\}$, with two directed edges between them. Subgraph $\{A, B, C, D\}$ has no incoming edges and can be colored on its own (*i. e.* without reference to the rest of graph G). Component $\{E, F, G\}$ has two incoming edges. Observe that these can be thought of as, in the worst case, reducing by two the number of colors available in our palette $\mathcal{D}$ when coloring $\{E, F, G\}$. Now, $\mathcal{D}$ contains at least $\chi(G) = 4$

colors (since we assume coloring of graph G is feasible) while subgraph $\{E, F, G\}$ is colorable using only two colors. Hence, regardless of the colors of vertices C and D on the two incoming edges, sufficient colors are always available to color subgraph $\{E, F, G\}$. We formalize these observations in the following theorem.

Definition 6 (Subgraph of G' generated by V_i)**.** The subgraph of graph $G' = (V, \mathcal{C})$ generated by V_i is the graph $(V_i, \{(i, j) : i, j \in V_i, (i, j) \in \mathcal{C}\})$. That is, the graph with V_i as its vertex set and with all the arcs in G' that have both their endpoints in V_i. With a slight abuse of notation, we will identify the subgraph with the vertex set V_i that generates it.

Definition 7 (In-degree of a subgraph)**.** The in-degree of the subgraph graph $G' = (V, \mathcal{C})$ generated by V_i, denoted by $deg(V_i)$, is the number of vertices $j \in V \setminus V_i$ that have at least one edge $(j, k) \in \mathcal{C}$, $k \in V_i$ ending in V_i.

Theorem 2. *Let $V = \bigcup_{i=1}^{p} V_i$, $V_i \bigcap V_j = \emptyset$ be a partition of the vertex set V of oriented graph $G' = (V, \mathcal{C})$ such that (i) the subgraph generated by V_i, $i \in \{1, \ldots, p\}$ is strongly connected and (ii) the subgraph generated by the union $\cup_{i \in S} V_i$ of any subset $S \subset \{1, \ldots, p\}$ is not strongly connected. That is, directed edges may exist between strongly connected components, but their union is not strongly connected. Let D be the number of colors available in our palette $\mathcal{D}$ and let $\chi(V_i)$ be the chromatic number of the (undirected) subgraph of $G = (V, \mathcal{M})$ generated by V_i. Suppose that*

$$\chi(V_i) \leq D - deg(V_i),\ i = 1, \ldots, p$$

Then for any coloring problem that admits a proper coloring and that fulfills condition (B) of Theorem 1, Algorithm 1 will find a proper coloring in almost surely finite time.

Proof. The original coloring problem is satisfiable by hypothesis so we have at least $\chi(G)$ available colors D in our palette, where $\chi(G)$ is the chromatic number of graph G. We need to consider two cases. Case 1: $deg(V_i) = 0$. Since $\chi(V_i) \leq \chi(G)$ (since V_i is a subgraph of G), at least one proper coloring of subgraph V_i exists and we can Theorem 1 to establish that Algorithm 1 will almost surely find a proper coloring. Case 2: $deg(V_i) > 0$. The incoming edges reduce by at most $deg(V_i)$ the choice of the colors available for subgraph V_i. Hence, provided $\chi(V_i) \leq D - deg(V_i)$ then we can apply Theorem 1 to subgraph V_i in isolation from the rest of graph G to establish that Algorithm 1 will almost surely find a proper coloring. □

6 Performance on Random Graphs

The upper bound in Theorem 1 is a worst case bound, and in addition we believe that it may not be tight. Hence, it is important to also evaluate the performance of Algorithm 1 using numerical measurements. In this section we present measurements for a class of random graphs that are based on an idealised model of wireless network interference. These graphs have been widely studied [7] and provide a method for technology-neutral evaluation. In Section 7 we evaluate performance in a technology specific manner using graphs derived from the WiGLE database of 802.11 hot spot locations.

6.1 Random Graph Model

We use realizations drawn from the Directed Boolean Model (DBM) described in [7]. The vertices of the graph are drawn from a Poisson point process in $[0,1]^2$ with intensity λ (with appropriate re-scaling to a required area – in the examples here we rescale to an area of $100\,m^2$). In the original undirected Boolean model (also known as the blob model [see 11, Section 10.5]), each vertex is the center of a closed ball of random radius. The radii of the balls are independently and identically distributed. The (undirected) connectivity graph is obtained by adding an edge between all pairs of points whose balls overlap, *i. e.* $B(x) \cap B(y) = \emptyset$, where $B(x), B(y)$ denote the balls centered on vertices x, y respectively. To obtain a directed graph, following [7] we slightly change the above rule and put a directed edge between x and y if $y \in B(x)$ and an edge between y and x if $x \in B(y)$. This modified model is referred to as the Directed Boolean Model (DBM).

In our measurements the radii are chosen uniformly at random from the finite set of the coverage areas corresponding to transmitting powers in the range $12-20\,\mathrm{dBm}$, with steps of $2\,\mathrm{dBm}$, and a specified detection threshold R. We use the 3GPP path loss model for indoor environments [2], based on the Okumura-Hata log-distance model

$$PL_{dB}(d) = 43.3 \cdot \log_{10} d + 11.5 + 20 \cdot \log_{10} f$$

where d is the distance in meters and f is the frequency in GHz. In the examples here we select fixed frequency $f = 2.412\,\mathrm{GHz}$. For detection threshold R in dB and transmit power P in dB, the coverage radius is then given by

$$d : PL_{dB}(d) + P \geq R$$

Figures 2 and 4 show examples of graph generated using this model.

We focus in the most challenging cases by selecting the number D of available colors equal to the minimum feasible value $\chi(G)$.

6.2 Meeting Connectivity Requirements

Theorems 1 and 2 place connectivity requirements on the induced sensing graph G' in order to ensure that Algorithm 1 converges to a satisfying assignment. We begin by evaluating the fraction of random graphs in the Directed Boolean Model that meet these requirements. Figure 3 plots this fraction for a range of detection thresholds R and vertex densities λ. It can be seen that for detection thresholds below $-15\,\mathrm{dBm}$ greater than 96% of graphs satisfy the connectivity

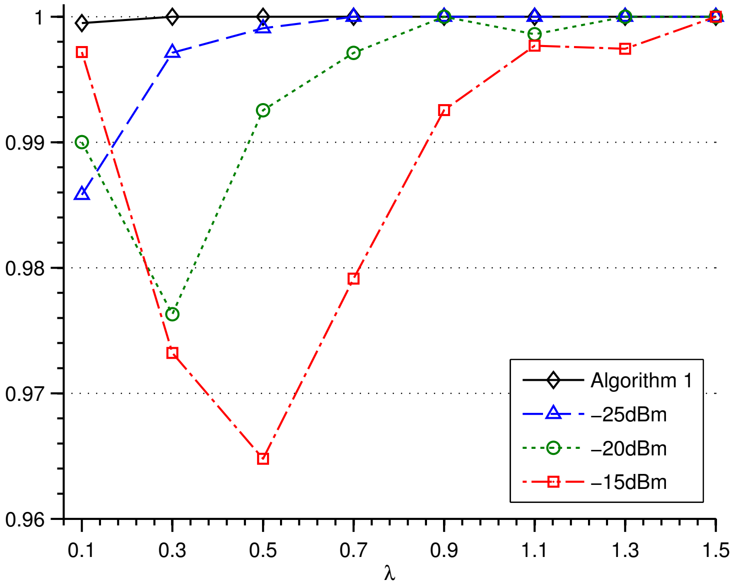

Figure 3: Fraction of DBM graphs nodes satisfying connectivity requirements of Theorem 2 versus the detection threshold. Additionally, the fraction of nodes correctly colored by Algorithm 1 for detection threshold of -25 dBm is shown.

requirements. Figure 4 shows some examples of some DBM graphs corresponding to a -25 dBm threshold. Observe that they consist of a number of connected components and so the relaxed connectivity conditions provided by Theorems 2 are of considerable importance here. Note also that modern wireless devices typically have a noise floor of less than -70 dBm and so -25 dBm is conservative.

Also shown in Figure 3 is the measured fraction of vertices for which Algorithm 1 successfully found a satisfying assignment for a detection threshold of -25 dBm. It can be seen that greater than 99.9% of vertices are successfully colored by the algorithm. For $\lambda = 0.5$ and detection threshold of -15 dBm, 0.04% of the vertices that does not fulfill the conditions of Theorem 2 are still correctly colored by Algorithm 1. This small gap can be explained with the fact that the conditions of Theorem 2 are sufficient, but not necessary for convergence: some topologies can lead to convergence for their particular structure or because of a fortunate initial condition (see Figure 4 for some examples).

6.3 Convergence Rate

Figure 5 shows the measured distribution of convergence time for Algorithm 1 versus the detection threshold used for sensing. For a threshold of -25 dBm, the mean convergence time is less than 2000 iterations. When the required

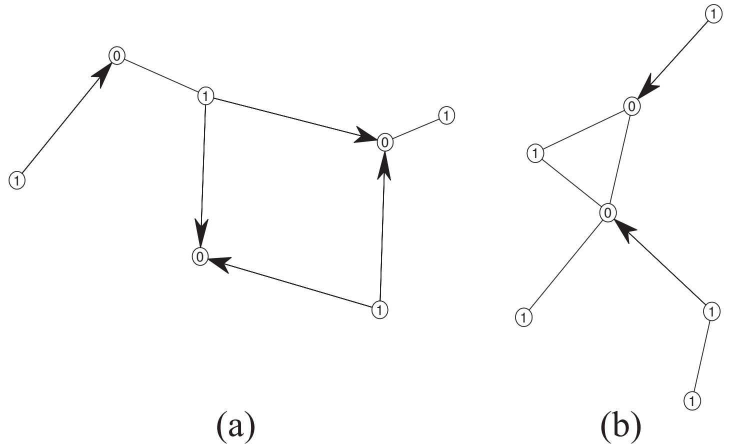

Figure 4: Example DBM graphs. The nodes labeled with 1 are the one that satisfy the connectivity conditions of Theorem 2

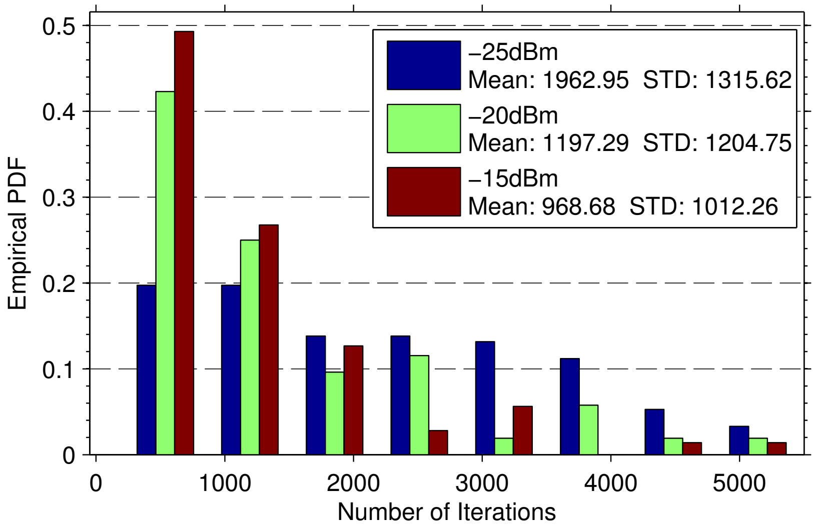

Figure 5: Measured convergence rate of Algorithm 1 for DBM graphs using a number of available colors equal to the chromatic number χ of the graph for three different detection thresholds. The density is $\lambda = 0.5$.

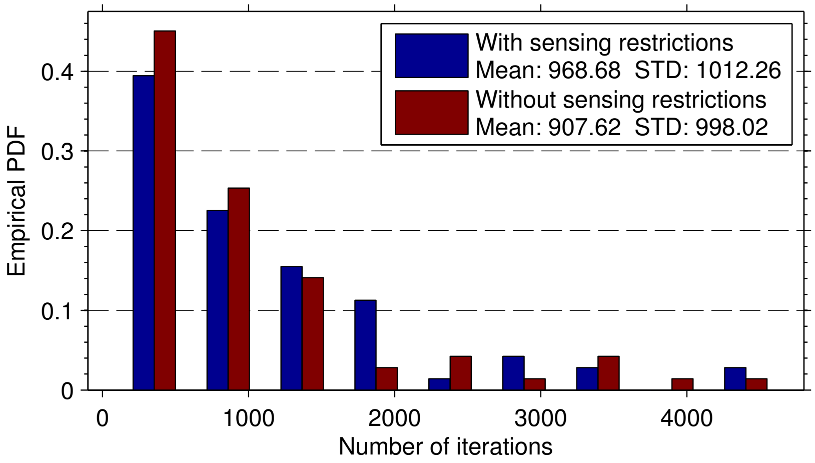

Figure 6: Measured convergence rate of Algorithm 1 for DBM graphs with detection threshold of $-15\,\mathrm{dBm}$ and density of $\lambda = 0.5$, with and without sensing restrictions.

threshold is increased to $-15\,\mathrm{dBm}$, the mean convergence time decreases to less than 1000 iterations. These measurements are for a link density of $\lambda = 0.5$, corresponding to on average 50 wireless links in an area of $100m^2$. Recall that we selected the number D of available colors equal to the minimum feasible $\chi(G)$, thereby focussing on the most challenging situations. For larger numbers of colors it can be verified that the convergence time decreases exponentially in the number of colors above $\chi(G)$.

The comparison of the bounds given by Theorem 1 with the case without sensing restrictions given by Corollary 2 suggests that sensing restrictions lead to an increase in the convergence time. This is indeed the case, as shown in Figure 6, where the convergence rate of Algorithm 1 is shown with and without sensing restrictions for DBM graphs with $\lambda = 0.5$ and detection threshold of $-15\,\mathrm{dBm}$. However for DMB graphs it can be seen that this increase is small. We also analyzed in section 7.1 the impact of the number of available colors on the convergence time.

7 Case Study: Manhattan WiFi Hots Spots

From the online database WiGLE [3] we obtained the locations of WiFi wireless Access Points (APs) in an approximately $150\,m^2$ area at the junction of 5th Avenue and 59th Street in Manhattan[1]. This space contains 81 APs utilizing the IEEE 802.11 wireless standard. We model radio path loss with distance as

[1] The extracted (x,y,z) coordinate data used is available online at www.hamilton.ie/net/xyz.txt

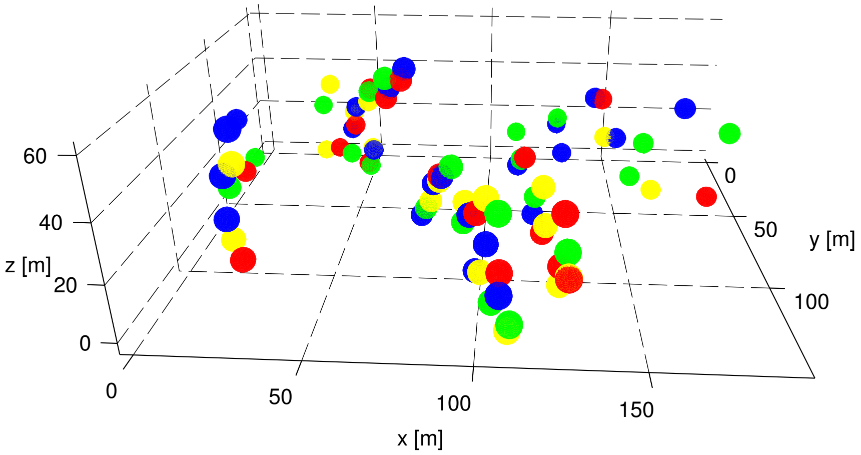

Figure 7: Example assignment for Manhattan WiFi hot spots. Wireless access points are indicated by points and the color indicates the radio channel selected by the AP.

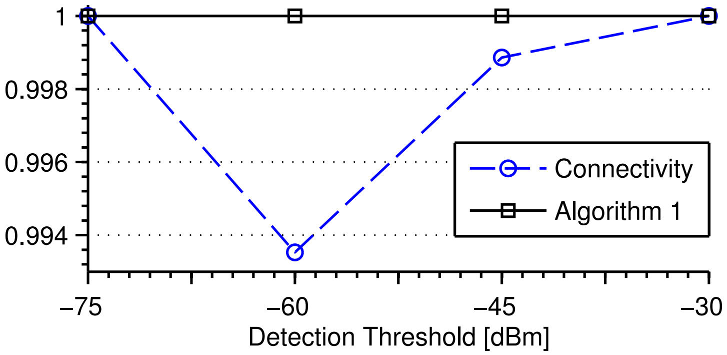

Figure 8: Connectivity for Manhattan WiFi hot spots. Each measure is the fraction of nodes that satisfy connectivity requirements of Theorem 2. The fraction of nodes correctly colored by Algorithm 1 is also shown.

d^α, where d is the distance in meters and $\alpha = 4.3$ is the path loss exponent (consistent with the 3GPP indoor propagation model [2]), and the AP transmit powers are selected uniformly at random in the range $12 - 20\,$dBm, with steps of 2 dBm. The aim of each AP is to select its radio channel in such a way as to ensure that it is sufficiently different from nearby WLANs. This can be written as a coloring problem with $N = 81$ APs and N variables x_i corresponding to the channel of AP i, $i = 1, \dots, N$. As per the 802.11 standard [1] and FCC regulations, each AP can select from one of 11 radio channels in the 2.4 GHz band and so the x_i, $i = 1, 2, \dots, N$ take values in $D = \{1, 2, \dots, 11\}$. To avoid excessive interference each AP requires that the received signal strength from other APs sharing the same channel is attenuated by at least $-60\,$dB. When all APs use the maximum transmit power of 18 dBm allowed by the 802.11 standard, this requirement is met when the received power is less than $-45\,$dBm and ensures that the SINR is greater than 20 dB (sufficient to sustain a data rate of 54 Mbps when the connection is line of sight and channel noise is Gaussian [10]).

The APs do not belong to a single administrative domain and so a decentralised solver is required. The presence of hidden terminals means that the solver must find a satisfying solution while subject to sensing asymmetry.

The connectivity requirement of Theorem 2 was observed to be satisfied $> 99\%$ of examples, see Figure 8.

7.1 Convergence Time

Algorithm 1 was observed to converge in less than 1000 iterations in all examples. Figure 9 shows the measured distribution of convergence time for Algorithm 1 versus the detection threshold used for sensing. For a threshold of $-45\,$dBm, corresponding to the target requirement noted above, the mean convergence

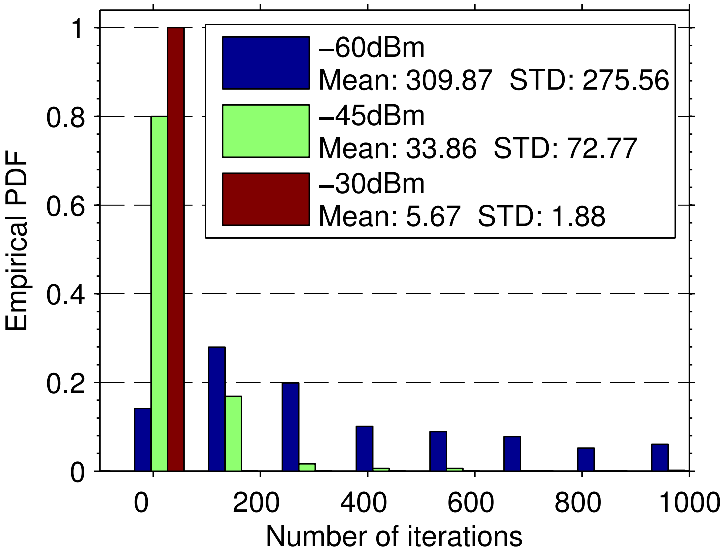

Figure 9: Measured convergence rate of Algorithm 1 for Manhattan WiFi hot spots using a number of available colors equal to the chromatic number χ of the graph.

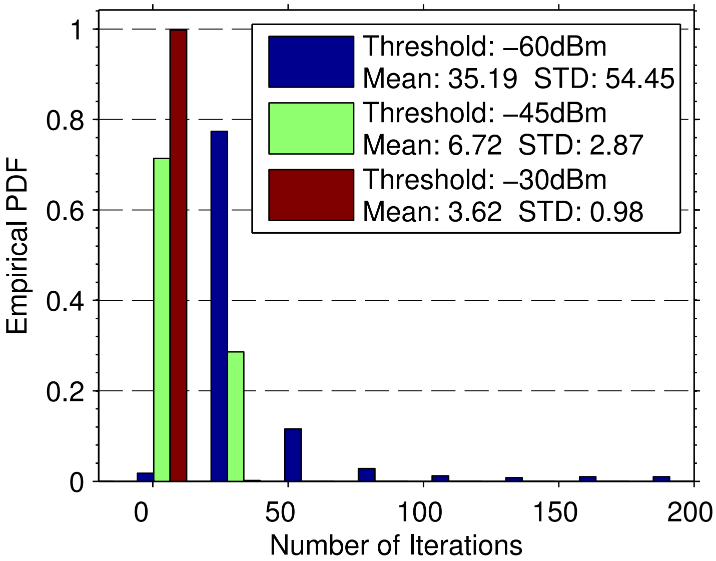

Figure 10: Measured convergence rate of Algorithm 1 for Manhattan WiFi hot spots using a number of available colors equal to $\chi+2$, where χ is the chromatic number of the graph.

time is less than 34 iterations. In a prototype lab set-up we have shown that an update interval of less than 10 seconds is feasible on current 802.11 hardware. Thus the mean time to convergence is under 6 minutes, which is a reasonable time-frame for practical purposes. When the required threshold is increased to $-30\,\text{dBm}$, the mean convergence time decreases to less than 6 iterations *i. e.* under 1 minute when each iteration takes 10 seconds.

To examine the impact of the number of available colors we compare, in Figures 9 and 10, the convergence time when the number of colors is equal to χ and $\chi+2$ respectively. For a detection threshold of $-60\,\text{dBm}$, adding two colors reduces the mean convergence time of almost 10 times.

8 Conclusions

In this paper we focus on graph-coloring problems, a subset of general CSPs. We constructively establish the existence of decentralized learning-based solvers that are able to find satisfying assignments even in the presence of sensing restrictions, in particular sensing asymmetry of the type encountered when hidden terminals are present. Our main analytic contribution is to establish sufficient conditions on the sensing behaviour to ensure that the solvers find satisfying assignments with probability one. These conditions take the form of connectivity requirements on the induced sensing graph. These requirements are mild, and we demonstrate that they are commonly satisfied in wireless allocation tasks. We explore the impact of sensing constraints on the speed which a satisfying assignment is found, showing the increase in convergence time is not significant in common scenarios.

Our results are of considerable practical importance in view of the prevalence of both communication and sensing restrictions in wireless resource allocation problems. The class of algorithms analysed here requires no message-passing whatsoever between wireless devices, and we show that they continue to perform well even when devices are only able to carry out constrained sensing of the surrounding radio environment.

Appendix: Proofs

Consider graph $G' = (V, \mathcal{C})$. Let

$$A = \bigcup \{\vec{x} \colon \Phi_m(\vec{x}) = 1 \text{ for all } m \in \mathcal{C}\},$$

denote the set of assignments which are absorbing for Algorithm 1 and

$$B = \bigcup \{\vec{x} \colon x_i \neq x_j \text{ for all } i \leftrightarrow j\},$$

the set of proper colorings, with $A \supseteq B$. Under condition (A) of Theorem 1, $A = B$ and all absorbing assignments are also satisfying. When the coloring problem is feasible then $A \neq \emptyset$ (at least one satisfying assignment exists). Let

$a \in A$ be a target satisfying assignment. We will refer to the assignment at time step t as $\vec{x}(t)$. Let $F_{\vec{x}(t)}$ denote the set of vertices that have their target color, *i. e.* $F_{\vec{x}(t)} = \{i \colon i \in V, x_i(t) = a_i\}$. Furthermore, let $U_{\vec{x}(t)}$ denote the set of unsatisfied vertices, *i. e.* $U_{\vec{x}(t)} = \{i \colon i, j \in V, x_j(t) = x_i(t), j \to i\}$, where $i \to j$ and $j \leftarrow i$ denote the existence of an oriented edge $(i,j) \in \mathcal{C}$. Define $\gamma = \min(a,b)/(D-1+a/b)$.

Lemma 2. If a vertex is unsatisfied, when using Algorithm 1 the probability that the vertex chooses any color j at the next step is greater than or equal to γ.

Proof. This follows from step 5 of Algorithm 1. □

Lemma 3. Given any satisfiable CP and an information set $\{\mathcal{C}_1, \dots, \mathcal{C}_N\}$ with starting unsatisfied assignment $\vec{x}(0) \in D^N, \vec{x}(0) \notin A$ such that $F_{\vec{x}(0)} \not\supseteq U_{\vec{x}(0)}$, Algorithm 1 will reach an assignment $\vec{x}(\tilde{t})$ such that $F_{\vec{x}(\tilde{t})} \supsetneq F_{\vec{x}(0)}$ and $F_{\vec{x}(\tilde{t})} \supseteq U_{\vec{x}(\tilde{t})}$ in $\tilde{t} \leq |F_{\vec{x}(\tilde{t})}| - |F_{\vec{x}(1)}| \leq N$ steps with probability greater than

$$\gamma^{\sum_{k=|F_{\vec{x}(0)}|}^{|F_{\vec{x}(\tilde{t})}|} k} < \gamma^{N(N+1)/2}$$

In other words, all vertices that had their target color in $\vec{x}(0)$ will still have it in $\vec{x}(\tilde{t})$, and all unsatisfied vertices in $\vec{x}(\tilde{t})$ will have their target color.

Proof. At the first step we consider the event that changes the assignment to

$$x_i(1) = \begin{cases} a_i & \text{if } i \in U_{\vec{x}(0)}, \\ x_i(0) & \text{otherwise.} \end{cases} \tag{4}$$

This event is feasible since Algorithm 1 ensures that all satisfied vertices will remain unchanged and each unsatisfied vertex may change its color. The probability that this event happens is greater than $\gamma^{|U_{\vec{x}(0)}|}$. After this step we have $F_{\vec{x}(1)} = F_{\vec{x}(0)} \cup U_{\vec{x}(0)}$. Now, the set of unsatisfied variables $U_{\vec{x}(1)}$ could have changed. If $U_{\vec{x}(1)} \subseteq F_{\vec{x}(1)}$, we have finished, otherwise we consider again the event that changes the assignment similarly to equation (4), *i. e.* at generic step t we have

$$x_i(t) = \begin{cases} a_i & \text{if } i \in U_{\vec{x}(t-1)}, \\ x_i(t-1) & \text{otherwise.} \end{cases}$$

The probability of this happening is greater than $\gamma^{|U_{\vec{x}(t-1)}|}$, and it can be lower bounded by $\gamma^{|F_{\vec{x}(t)}|}$ because $F_{\vec{x}(t)} = F_{\vec{x}(t-1)} \cup U_{\vec{x}(t-1)}$. Since while $U_{\vec{x}(t-1)} \not\subseteq F_{\vec{x}(t-1)}$ we have $F_{\vec{x}(t)}$ is a strictly growing set, and we have a finite number of vertices N, a finite time $\tilde{t} \leq N$ exists after which we will necessarily have $U_{\vec{x}(\tilde{t})} \subseteq F_{\vec{x}(\tilde{t})}$. The worst case in regards to the number of steps is when at each step, only one new vertex is added to $F_{x(t)}$, giving us the bound for the number of steps of $|F_{x(\tilde{t})}| - |F_{x(1)}|$. □

Lemma 4. Consider any satisfiable CP and an information set $\{\mathcal{C}_1, \dots, \mathcal{C}_N\}$ with induced graph $G' = (V, \mathcal{C})$ and color $x_i(t) \in D$ associated with each vertex $i \in V$ at time t. Let $A \subset D^{|V|}$ denote the set of satisfying assignments. Suppose $|V| > 1$, $\vec{x}(0) \notin A$ (the initial choice of colors is not a satisfying assignment) and graph G' is strongly connected. Let $a \in A$ be an arbitrary satisfying assignment. If $F_{\vec{x}(0)} \supseteq U_{\vec{x}(0)}$, there exists a pair of vertices k, j with same color such that $j \to k$ and $k \not\to j$, with $k \in U_{\vec{x}(0)}$ and $j \notin U_{\vec{x}(0)}, j \notin F_{\vec{x}(0)}$; in other words, a vertex k exists that is unsatisfied by a satisfied vertex j that doesn't have final color, and the minimum path length between k and j is greater than 1.

Proof. Consider any unsatisfied vertex $k \in U_{\vec{x}(0)}$. At least one such vertex exists because $\vec{x}(0) \notin A$. The hypothesis $F_{\vec{x}(0)} \supseteq U_{\vec{x}(0)}$ ensures $x_k(0) = a_k$. Since k is unsatisfied, there exists a node j such that $j \to k$ and $x_j(0) = x_k(0)$, and $x_j(0) \neq a_j$, because $x_k(0) = a_k$, $x_j(0) = x_k(0) = a_k$ and since $k \leftrightarrow j$ in G, we must have $a_j \neq a_k$. The hypothesis $F_{\vec{x}(0)} \supseteq U_{\vec{x}(0)}$ also ensures that j is satisfied, because if it was unsatisfied it should have its final color because $F_{\vec{x}(0)} \supseteq U_{\vec{x}(0)}$ and this would contradict the property just proved that $x_j(0) = a_j$. Since j is satisfied and it has same color than k, we have $k \not\to j$. □

Definition 8 (1-rotation)**.** A *1-rotation* is an operator P acting on vector $\vec{s} = (s_1, s_2, \dots, s_m)$, $m > 1$, such that $P(\vec{s})_i = s_{i+1}$, $i = \{1, 2, \dots, m-1\}$ and $P(\vec{s})_m = s_1$. Repeating a 1-rotation m times yields the identity operation, *i. e.* $P^m(\vec{s}) = \vec{s}$.

Lemma 5. Consider any satisfiable CP and an information set $\{\mathcal{C}_1, \dots, \mathcal{C}_N\}$ and induced graph $G' = (V, \mathcal{C})$ and color $x_i(t) \in D$ associated with each vertex $i \in V$ at time t. Suppose there exists a cycle $p_1 \to p_2 \to \cdots \to p_m \to p_{m+1} \to p_1 \subseteq G'$, $m > 1$, with $x_{p_{m+1}}(0) = x_{p_1}(0)$ at time $t = 0$. Let $\vec{s}(0) = (x_{p_1}(0), x_{p_2}(0), \dots, x_{p_m}(0))$. With probability greater than γ^{Nm}, after m time steps Algorithm 1 will realize a 1-rotation of the vector $\vec{s}(0)$, *i. e.* $\vec{s}(m) = P(\vec{s}(0)) = (x_{p_2}(0), x_{p_3}(0), \dots, x_{p_m}(0), x_{p_1}(0))$, while leaving the colors of all other vertices unchanged.

Proof. Observe that at time $t = 0$ vertex p_1 is unsatisfied since $x_{p_{m+1}}(0) = x_{p_1}(0)$ and $p_{m+1} \to p_1$. Consider the event that at time $t = 1$

$$\begin{aligned} x_{p_1}(1) &= x_{p_2}(0) \\ x_{p_2}(1) &= x_{p_2}(0) \\ &\vdots \\ x_{p_m}(1) &= x_{p_m}(0) \end{aligned}$$

and the colors of all other vertices remain unchanged. This event is feasible since Algorithm 1 ensures that all satisfied vertices will remain unchanged and each unsatisfied vertex may choose any color from set D with probability at least γ. From the latter, the event described occurs with probability greater than γ^N.

Observing that vertex p_2 is now unsatisfied since $x_{p_1}(1) = x_{p_2}(0) = x_{p_2}(1)$ and $p_1 \to p_2$, suppose that at time $t = 2$

$$\begin{aligned} x_{p_1}(2) &= x_{p_1}(1) = x_{p_2}(0) \\ x_{p_2}(2) &= x_{p_3}(1) = x_{p_3}(0) \\ x_{p_3}(2) &= x_{p_3}(1) \\ &\vdots \\ x_{p_m}(2) &= x_{p_m}(1). \end{aligned}$$

Again this event is feasible and occurs with probability greater than γ^N. After m such steps we have $\vec{s}(m) = P(\vec{s}(0))$ as claimed, and this sequence of events will occur with probability greater than γ^{Nm}. □

Lemma 6. Consider any satisfiable CP and an information set $\{\mathcal{C}_1, \ldots, \mathcal{C}_N\}$ with induced graph $G' = (V, \mathcal{C})$ and color $x_i(t) \in D$ associated with each vertex $i \in V$ at time t. Let $A \subset D^{|V|}$ denote the set of satisfying assignment. Suppose $\vec{x}(0) \notin A$ (the initial choice of colors is not a satisfying assignment) and graph G' is strongly connected. Let $d \in D$ be an arbitrary color. Let $k \in V$ be an unsatisfied vertex and let j be a vertex such that $j \to k$, $x_j(0) = x_k(0)$ (at least one such vertex exists since k is unsatisfied). With probability greater than γ^{N^3}, in $\tilde{t} < N^2$ steps Algorithm 1 will choose $\vec{x}(\tilde{t})$, such that $x_i(\tilde{t}) = x_i(0)\ \forall i \in \{i : i \in V, i \neq j\}$ and $x_j(\tilde{t}) = d$.

Proof. Since G' is strongly connected, there exists a cycle $k \to \cdots \to j \to k \subseteq G'$. Let us relabel the $m+1 > 1$ vertices in the cycle using the ordering induced by the cycle, *i. e.* $p_1 = k, p_{m+1} = j$ and so $p_1 \to p_2 \to \cdots \to p_{m+1} \to p_1$. Define vector $\vec{s}(t) = (x_{p_1}(t), \ldots, x_{p_{m-1}}(t), x_{p_m}(t))$. We need to consider two cases. $m = 1$. In this case the cycle is $p_1 \to p_2 \to p_1$. By assumption, $x_{p_2}(0) = x_{p_1}(0)$ and so vertex p_2 is unsatisfied since $p_1 \to p_2$. It follows that, with probability at least γ^N, after 1 time step Algorithm 1 will realize the event that vertex p_2 selects color d and the color of all other vertices remains unchanged. $m > 1$. Using Lemma 5, with probability greater than γ^{Nm} in m steps Algorithm 1 will realize a 1-rotation of the vector $\vec{s}(0)$ *i. e.* $\vec{s}(m) = (x_{p_2}(0), \ldots, x_{p_m}(0), x_{p_1}(0))$ leaving the colors of all other vertices unchanged. Observe that vertex $j = p_{m+1}$ must now be unsatisfied because $x_{p_m}(m) = x_{p_1}(0)$, $x_{p_{m+1}}(m) = x_{p_{m+1}}(0) = x_{p_1}(0)$ and $p_m \to p_{m+1}$. Now consider the event at time $m+1$ where vertex p_{m+1} takes the color of vertex p_1 (and the color of all other vertices remains unchanged). This event occurs with probability greater than γ^N. After $m+1$ steps we have $\vec{s}(m+1) = (x_{p_2}(0), \ldots, x_{p_m}(0), x_{p_1}(0))$ and $x_{p_{m+1}}(m+1) = x_{p_2}(0)$, and this event occurs with probability greater than $\gamma^{N(m+1)}$. Applying again Lemma 5, after a 1-rotation and changing the color of unsatisfied vertex p_{m+1} we have $\vec{s}(2m+2) = (x_{p_3}(0), \ldots, x_{p_1}(0), x_{p_2}(0))$ and $x_{p_{m+1}}(2m+2) = x_{p_3}(0)$. This state is reached after $2(m+1)$ steps with probability greater than $\gamma^{2N(m+1)}$. Repeating, after $m(m+1)$ steps $\vec{s}(m^2+m) = (x_{p_1}(0), \ldots, x_{p_{m-1}}(0), x_{p_m}(0))$ and $x_{p_{m+1}}(m^2+m) = d$ (where at the very last step we select the color of unsatisfied vertex p_{m+1} to equal d rather than the color of p_1). This state is reached after $m(m+1)$ steps with probability greater than $\gamma^{Nm(m+1)}$. Since $m \leq N$, $m(m-1) < N^2$ steps and $\gamma^{Nm(m-1)} > \gamma^{N^3}$. □

Lemma 7. Consider any satisfiable CP and an information set $\{\mathcal{C}_1, \dots, \mathcal{C}_N\}$ with induced graph $G' = (V, \mathcal{C})$ and color $x_i(t) \in D$ associated with each vertex $i \in V$ at time t. Let $A \subset D^{|V|}$ denote the set of satisfying assignments. Suppose $|V| > 1$, $\vec{x}(0) \notin A$ (the initial choice of colors is not a satisfying assignment) and graph G' is strongly connected. Let $a \in A$ be an arbitrary satisfying assignment. If $F_{\vec{x}(0)} \supseteq U_{\vec{x}(0)}$ with probability greater than γ^{N^3}, in $\tilde{t} \leq N^2$ steps Algorithm 1 will reach an assignment $\vec{x}(\tilde{t})$ such that $F_{\vec{x}(\tilde{t})} \supsetneq F_{\vec{x}(0)}$ and $|F_{\vec{x}(\tilde{t})}| = |F_{\vec{x}(0)}| + 1$;

Proof. Lemma 4 ensures a pair i, j exists such that $x_k(0) = a_k, x_k(0) \in U_{\vec{x}(0)}$ and $x_j(0) = x_k(0)$. Lemma 6 ensures that, in less than N^2 steps, with probability greater than γ^{N^3}, Algorithm 1 will reach an assignment in which vertex j assumes color a_j and the colors of all other vertices are unchanged. □

Proof of Theorem 1. Consider Algorithm 1 starting from an assignment $\vec{x}(0)$. Select an arbitrary valid solution $a \in A$. Since the CP is satisfiable, we have that $A \neq \emptyset$. We will exhibit a sequence of events that, regardless of the initial configuration, leads to a satisfying assignment with a probability for which we find a lower bound. We consider the following sequence, divided in two phases:

```
$t \leftarrow 0$
**repeat**
  **if** $F_{\vec{x}(t)} \not\supseteq U_{\vec{x}(t)}$ **then**
    **Phase 1** Applying Lemma 3, after $\tilde{t} \leq N$ steps $F_{\vec{x}(t+\tilde{t})} \supseteq U_{\vec{x}(t+\tilde{t})}$ and $F_{\vec{x}(t+\tilde{t})} \supsetneq F_{\vec{x}(t)}$ (so $|F_{\vec{x}(t+\tilde{t})}| \geq |F_{\vec{x}(t)}| + 1$). This event happens with probability greater than $\gamma^{N^2}$.
    $t \leftarrow t + \tilde{t}$
  **end if**
  **if** $U_{\vec{x}(t)} \neq \emptyset$ **then**
    **Phase 2** We have $F_{\vec{x}(t)} \supseteq U_{\vec{x}(t)}$. Applying Lemma 7, after $\tilde{t} < N^2$ steps $|F_{\vec{x}(t+\tilde{t})}| = |F_{\vec{x}(t)}| + 1$. This event happens with probability greater than $\gamma^{N^3}$.
    $t \leftarrow t + \tilde{t}$
  **end if**
**until** $U_{\vec{x}(t)} = \emptyset$ .
```

This sequence is terminating, because the set $F_{\vec{x}(t)}$ is strictly increasing, and when $|F_{\vec{x}(t)}| = N$ we necessarily have $U_{\vec{x}(t)} = \emptyset$. Each vertex will be added to $F_{\vec{x}(t)}$ only once, either by Phase 1 or Phase 2.

When a vertex is added by Phase 1, it will require at most N steps and occur with probability at least $\gamma^{N(N+1)/2}$. When added by Phase 2, it will require at most N^2 steps and occur with probability at least γ^{N^3}. Since $N \leq N^2$ and $\gamma^{N(N+1)/2} \geq \gamma^{N^3}$ for $N > 1$, we can therefore upper bound the total number of steps by $N \cdot N^2 = N^3$ and lower bound the probability of the sequence by $(\gamma^{N^3})^N = \gamma^{N^4}$.

Due to the Markovian nature of Algorithm 1 and the independence of the probability of the above sequence on its initial conditions, if this sequence does not occur in N^3 iterations, it has the same probability of occurring in the next N^3 iterations. The probability of convergence in $k \cdot N^3$ steps is greater than $1-(1-\gamma^{N^4})^k$. For $1-(1-\gamma^{N^4})^k \geq 1-\epsilon$ we require $k \leq \frac{\log \epsilon}{\log(1-\gamma^{N^4})} \leq -\frac{\log \epsilon}{\gamma^{N^4}} = e^{N^4 \log(\gamma^{-1})} \log(\epsilon^{-1})$ □

Proof of Corollary 2. After running Phase 1 in the proof of Theorem 1 for the first time, we have $F_{\vec{x}(t+\tilde{t})} \supsetneq F_{\vec{x}(t)}$. If $U_{\vec{x}(t)} = \emptyset$ we have finished without running Phase 2. Otherwise we must run Phase 2. But in this case we have from Lemma 4 that $\mathcal{C} \neq \mathcal{M}$ (because there exists a pair of vertices i, j such that $j \to k$ and $k \not\to j$), leading to a contradiction. So after Phase 1 $U_{\vec{x}(t)} = \emptyset$ and Phase 2 is never executed. The running time of Phase 1 is no greater than N and occurs with probability at least $\gamma^{\sum_{k=1}^{N} k} = \gamma^{(N+1)N/2}$.

□

References

[1] IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications , Nov. 1997. P802.11, 1997.

[2] UE Radio transmission and Reception (FDD). Technical report, 3GPP TS25.101, 2004.

[3] wigle.net, 2010. URL `http://www.wigle.net/`.

[4] J. Barcelo, B. Bellalta, C. Cano, A. Sfairopoulou, M. Oliver, and K. Verma. Towards a collision-free WLAN: dynamic parameter adjustment in CSMA/E2CA. *EURASIP Journal on Wireless Communications and Networking*, 2011.

[5] A. Checco, R. Razavi, D.J. Leith, and H. Claussen. Self-Configuration of Scrambling Codes for WCDMA Small Cell Networks. 2012.

[6] P. Clifford and D.J. Leith. Channel dependent interference and decentralized colouring. *Network Control and Optimization*, pages 95–104, 2007.

[7] O. Dousse. Percolation in directed random geometric graphs. In *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, pages 601–605. IEEE, 2012.

[8] K.R. Duffy, C. Bordenave, and D.J. Leith. Decentralized constraint satisfaction. *CoRR*, abs/1103.3240, 2011.

[9] M. Fang, D. Malone, K.R. Duffy, and D.J. Leith. Decentralised learning macs for collision-free access in wlans. *Wireless Networks*, pages 1–16, 2010.

[10] A. Goldsmith. *Wireless communications.* Cambridge university press, 2005.

[11] G.R. Grimmett. *Percolation*, volume 321. Springer, 1999.

[12] S.T. Hedetniemi, D.P. Jacobs, and P.K. Srimani. Fault tolerant distributed coloring algorithms that stabilize in linear time. In *Proceedings of the IPDPS-2002 Workshop on Advances in Parallel and Distributed Computational Models*, pages 1–5, 2002.

[13] Ö. Johansson. Simple distributed $\Delta + 1$-coloring of graphs. *Information Processing Letters*, 70(5):229–232, 1999.

[14] B. Kauffmann, F. Baccelli, A. Chaintreau, K. Papagiannaki, C. Diot, et al. Self organization of interfering 802.11 wireless access networks. 2005.

[15] B. Kauffmann, F. Baccelli, A. Chaintreau, V. Mhatre, K. Papagiannaki, and C. Diot. Measurement-based self organization of interfering 802.11 wireless access networks. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1451–1459. IEEE, 2007.

[16] K. Kothapalli, M. Onus, C. Scheideler, and C. Schindelhauer. Distributed coloring in $\mathcal{O}(\sqrt{\log N})$-bits. In *Proc. of IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 2006.

[17] D.J. Leith, P. Clifford, V. Badarla, and D. Malone. WLAN channel selection without communication. *Computer Networks*, 2012.

[18] K.K. Leung and B.J. Kim. Frequency assignment for IEEE 802.11 wireless networks. In *sVehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 3, pages 1422–1426. IEEE, 2003.

[19] A. Mishra, V. Brik, S. Banerjee, A. Srinivasan, and W. Arbaugh. A client-driven approach for channel management in wireless LANs. In *IEEE Infocom*, volume 6, 2006.

[20] A. Mishra, V. Shrivastava, D. Agrawal, S. Banerjee, and S. Ganguly. Distributed channel management in uncoordinated wireless environments. In *Proceedings of the 12th annual international conference on Mobile computing and networking*, pages 170–181. ACM, 2006.

[21] L. Narayanan. *Handbook of Wireless Network and Mobile Computing*, chapter Channel assignment and graph multicoloring. Wiley Series on Parallel and Distributed Computing, 2002.

[22] A. Raniwala and T. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 2223–2234. IEEE, 2005.